Lab5

Ahmed Alhasan, Yash Pawal, Mohsen Pirmoradiyan

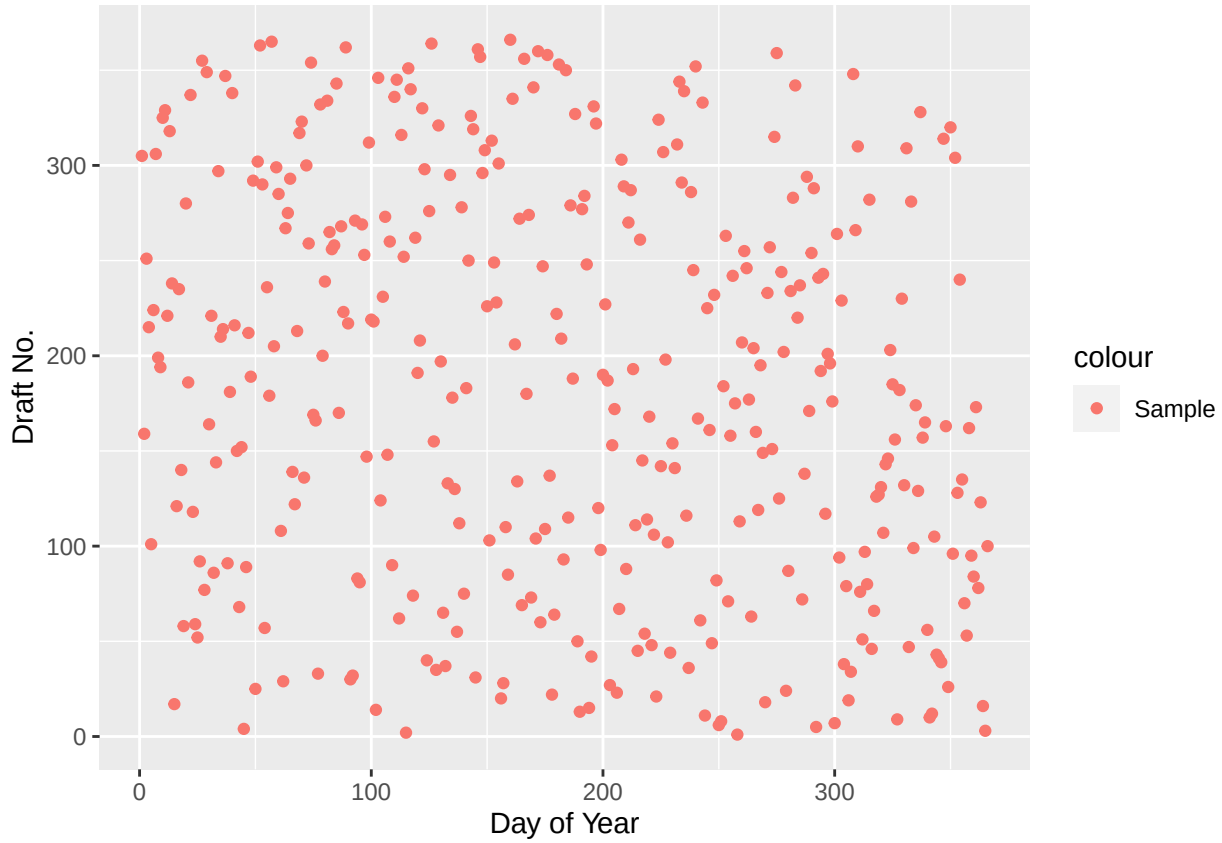
2/23/2020

Question 1: Hypothesis testing

```
data1 = read_xls("E:/LiU/2nd Semester/Computational Statistics/Labs/5/lottery.xls")
data1 = as.data.frame(data1)
X = data1$Day_of_year
Y = data1$Draft_No
df = data.frame(X =X, Y=Y)
```

1. Make a scatterplot of Y versus X and conclude whether the lottery looks random.

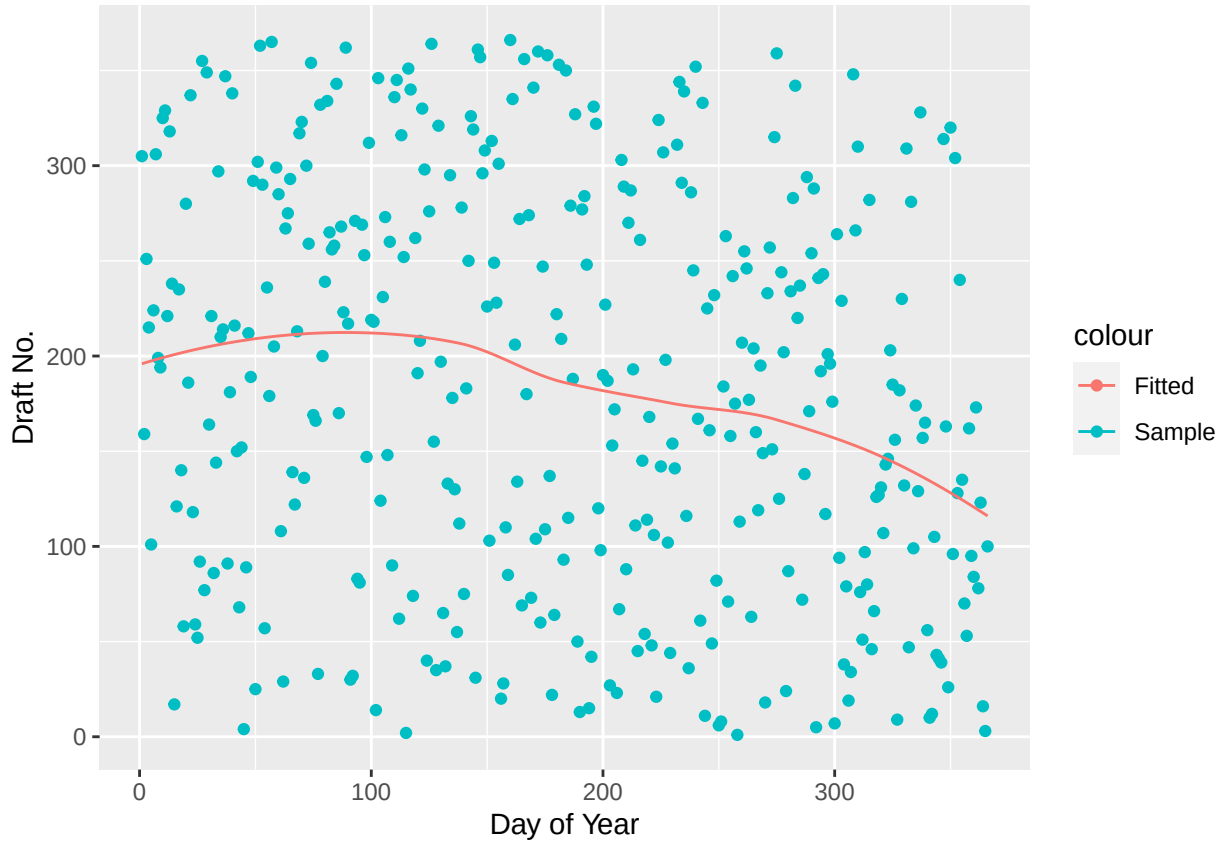
```
ggplot(df)+geom_point(mapping = aes(X,Y, col="Sample"))+
  xlab("Day of Year")+
  ylab("Draft No.")
```



The scatterplot shows no discernible pattern,thus the randomness may be concluded.

2. Compute an estimate $\hat{Y}$ of the expected response as a function of X by using a loess smoother.

```
loess.fit <- loess(Y~X, data = df)
Y.hat = loess.fit$fitted
df$Y.hat = Y.hat
ggplot(df)+geom_point(mapping = aes(X,Y, col="Sample"))+geom_line(mapping = aes(X,Y.hat, col="Fitted"))+
  xlab("Day of Year")+
  ylab("Draft No.")
```



It seems that this plot proves the randomeness as the sample points scatteres both sides of the fitted line randomely.

3. test statistics

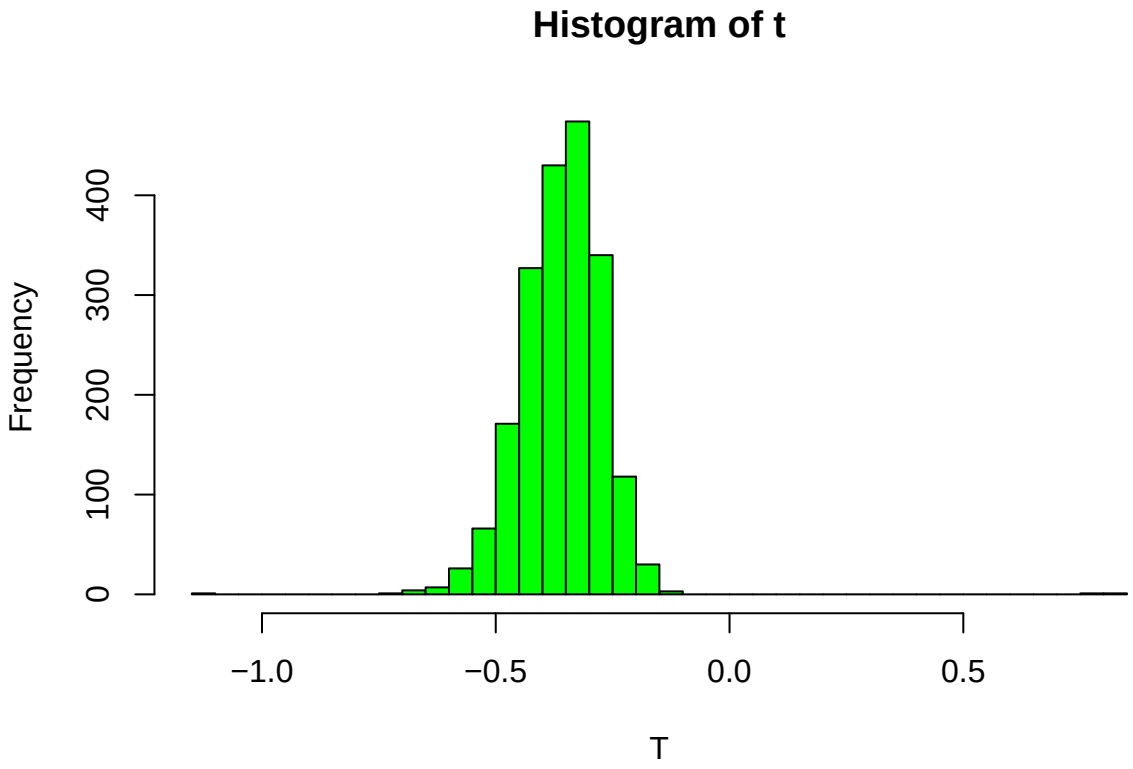
```
T = function(data1){

  data2 = data1
  fit <- loess(Y~X, data = data2)
  data2$pred = fit$fitted
  X.a = data2$X[which.min(data2$pred)]
  X.b = data2$X[which.max(data2$pred)]
  return((predict(fit, newdata = X.b)-predict(fit, newdata = X.a))/(X.b - X.a))

}
set.seed(12345)
```

```r
t= rep(0, 2000)
counter = 0
n= nrow(df)
for (b in 1:2000) {
  ind = sample(1:n, n, replace = TRUE)
  data1 = df[ind,]
  t[b] = T(data1)
  if(t[b] >0){counter = counter + 1}
}

hist(t, breaks = 40, xlab = "T", col = "Green")
```




```r
cat("P-value: ", counter/2000)
```

```
## P-value:  0.001
```

To implement nonparametric bootstrapping, 2000 times sampling with replacement were carried out on our data. In each iteration a loess model was fitted on the sampled data and the fitted value was computed. The maximum and minimum values of the fitted and the corresponding X values were obtained and used to calculate T. The histogrom above showes the result of 2000 run of this procedure. To test if the randomeness is proven by this test or not, we assume $H_0 : T > 0$ is our null hypothesis and $H_a : T \leq 0$ is the alternative. Under this null hypothesis if the p-value is statistically significant we can reject the null in favor of alternative. To calculate the p_value we averaged the T-values greater than 0 which resulted in 0.001. At a significance level of 0.05 we can reject the null hypothesis in favor of the alternative and conclude that there does not exist a discernible trend within the data,thus the randomeness may be proved.

4. permutation test

$$H_0 : \ Lottery\ is\ random$$

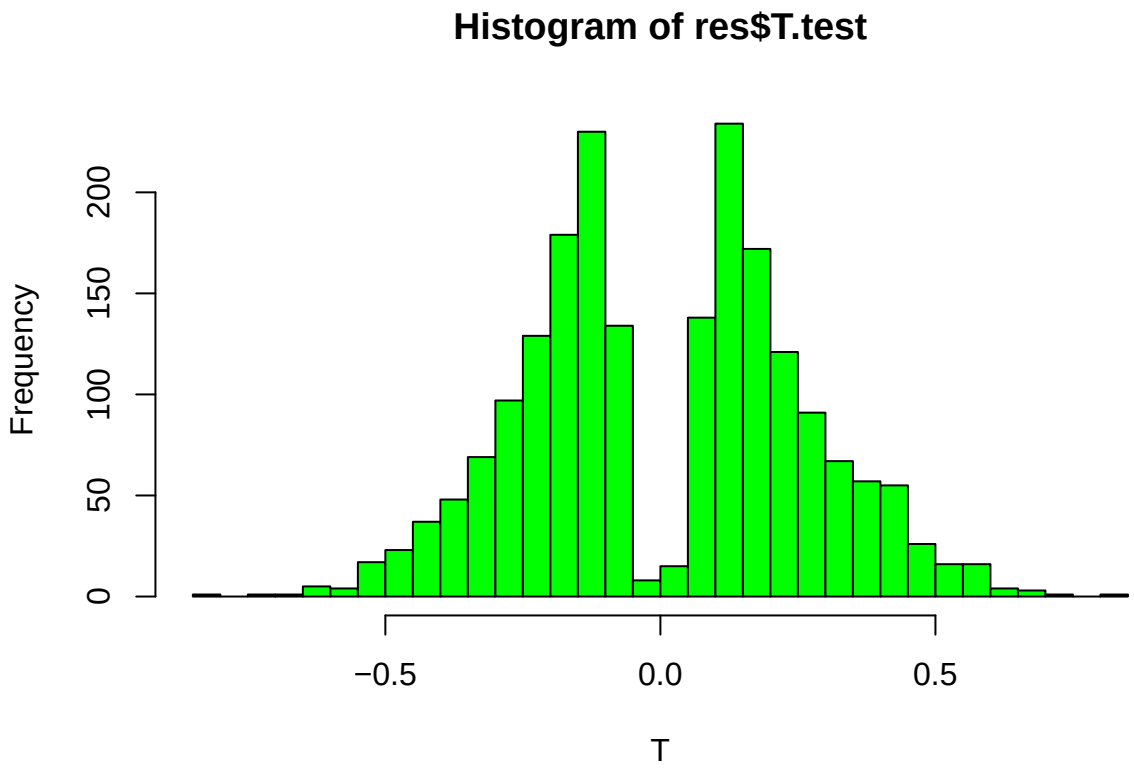
$$H_a : \ Lottery\ is\ non-random$$

```
permut <- function(data, B){
  data1$X = data$X
  t= rep(0, B)
  counter = 0
  for (b in 1:B) {
    data1$Y = sample(data$Y, length(data$Y), replace = FALSE)
    t[b] = T(data1)
    if(abs(t[b]) >= abs(T(df))){counter = counter + 1}

  }

  return(list(T.test=t,p.value=counter/B))
}

set.seed(12345)
res = permut(df, 2000)
hist(res$T.test, breaks = 40, col="green", xlab = "T")
```




```
res$p.value
```

```
## [1] 0.1595
```

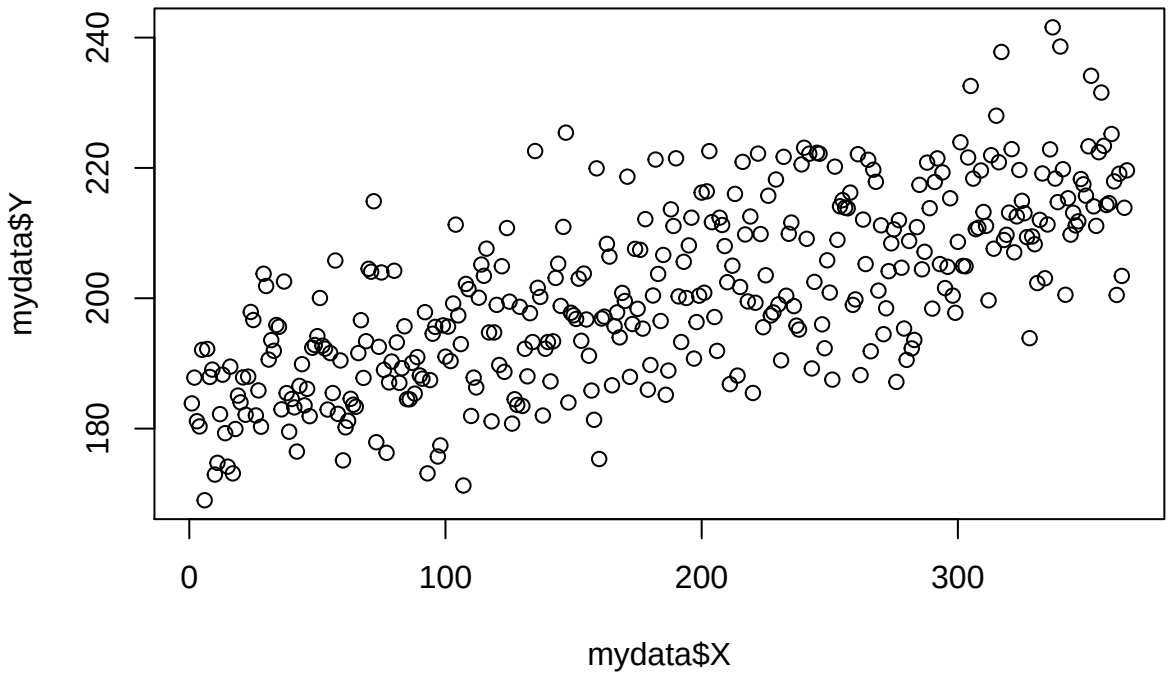
5. Make a crude estimate of the power of the test constructed in Step (4)

The data was generated. The scatter plot and the p-value are as follow:

```
n = nrow(df)

gen.data = function(n, alpha){
  newdata = data.frame(X=df$X, Y=0)
  for (i in 1:n) {
    beta = rnorm(1,183,10)
    newdata$Y[i] = max(0, min(alpha*newdata$X[i]+beta, 366))
  }
  return(newdata)
}

mydata=gen.data(n, 0.1)
plot(mydata$X, mydata$Y)
```



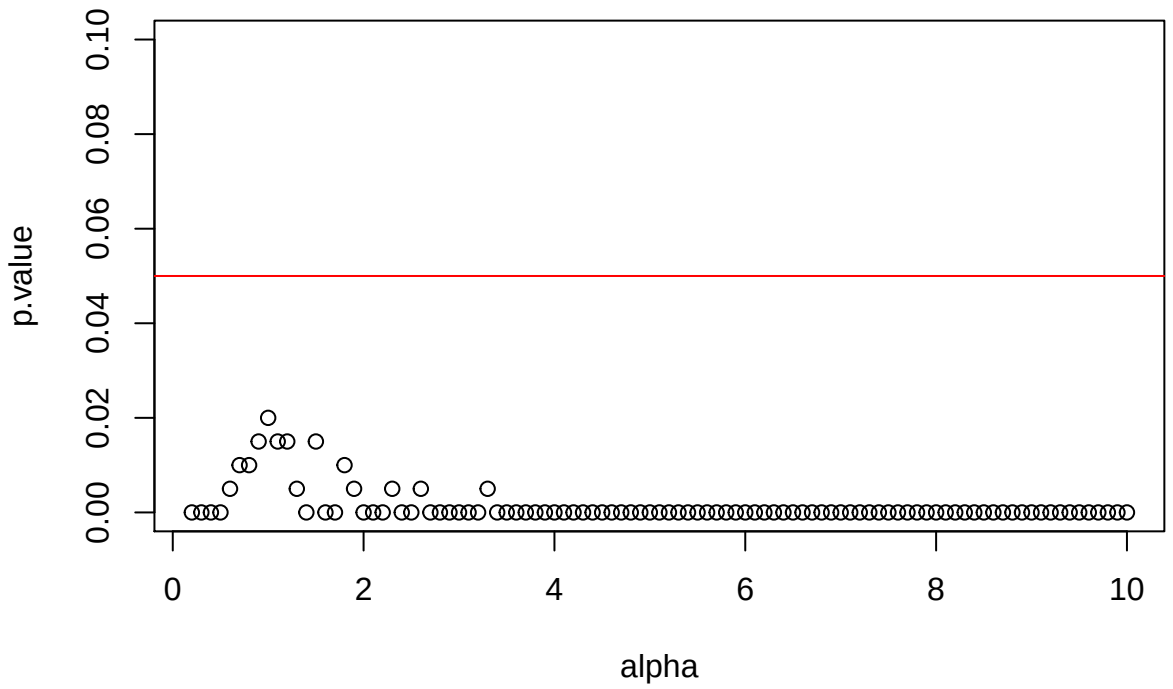

```
cat("The p-value: ",permut(mydata, 200)$p.value)
```

```
## The p-value:  0
```

The following graph shows the p-values vs. different values for α:

```
alpha = seq(0.2,10,0.1)
set.seed(12345)
p.value = sapply(1:length(alpha), function(i){
   permut(gen.data(n, alpha = alpha[i]), 200)$p.value})
```

```r
plot(alpha, p.value, ylim = c(0,0.1))
abline(a=0.05, b=0, col="red")
```



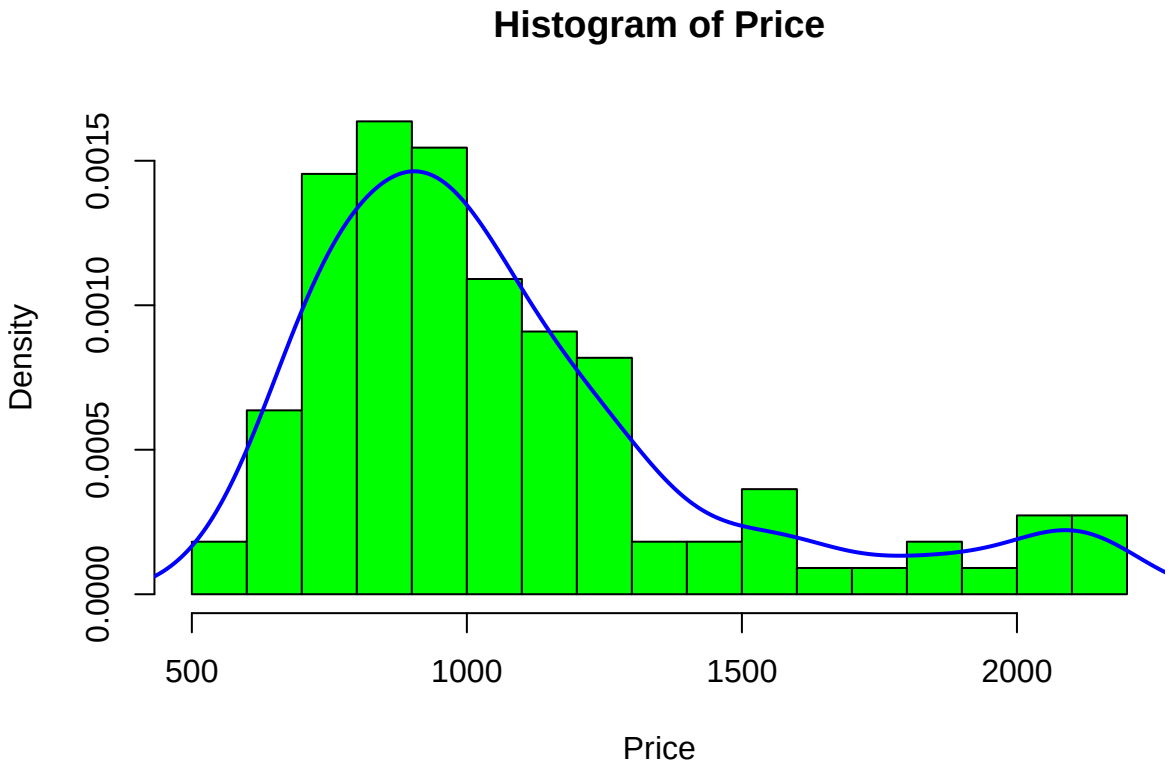
The response has been created through a linear relationship,hence the randomness is no longer valid in these datasets. As we expected the p.values all are less than 0.05, so resulted in rejection of null hypothesis(H_0 : *Lottery is random*).

Question 2: Bootstrap, jackknife and confidence intervals

1.Plot the histogram of Price. Does it remind any conventional distribution? Compute the mean price

```
price = read_xls("E:/LiU/2nd Semester/Computational Statistics/Labs/5/prices1.xls")
price = as.data.frame(price)

hist(price$Price,breaks = 20,prob=TRUE,col="Green", main = "Histogram of Price", xlab = "Price")
lines(density(price$Price), col="Blue", lwd=2)
```




```
cat("The mean value of the Price:\n", round(mean(price$Price),2))
```

```
## The mean value of the Price:
##  1080.47
```

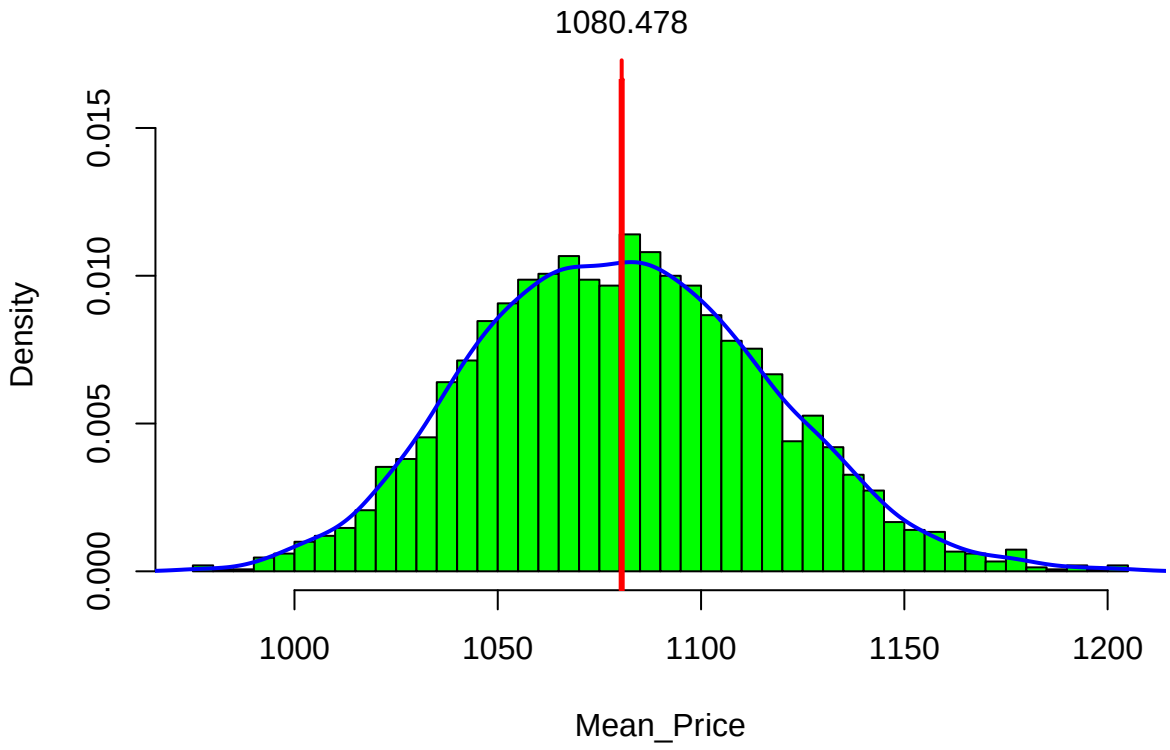
2.Estimate the distribution of the mean price of the house using bootstrap

We ran the bootstrap 3000 times and the following result were obtained:

```
f1 = function(data,id){
  data1 = data[id,]
  return(mean(data1$Price))

}
set.seed(12345)
res = boot(price, f1, R = 3000)
hist(res$t, breaks = 50, main = "", xlab = "Mean_Price", col = "Green",ylim = c(0,0.016), prob=TRUE)
```

```
lines(density(res$t), col="Blue", lwd=2)
abline(v=mean(res$t), col="red", lwd=3)
axis(3,at=mean(res$t),labels=round(mean(res$t),3), col.ticks="red", col="red", lwd=2)
```



The mean, variance, and the standard deviation of the mean distribution:

```
cat("The value of the mean: ",mean(res$t))
```

```
## The value of the mean:  1080.478
```

```
cat("\n\n")
```

```
cat("The value of the variance: ",var(res$t))
```

```
## The value of the variance:  1268.484
```

```
cat("\n\n")
```

```
cat("The value of the standard deviation: ",sd(res$t))
```

```
## The value of the standard deviation:  35.61579
```

Confidence Intervals

95% confidence interval for the mean price using bootstrap percentile

```
print(boot.ci(res, type = "perc"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 3000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   (1015, 1152 )
## Calculations and Intervals on Original Scale
```

95% confidence interval for the mean price using bootstrap BCa

```
boot.ci(res, type = "basic")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 3000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   (1009, 1146 )
## Calculations and Intervals on Original Scale
```

95% confidence interval for the mean price using first-order normal approximation

```
boot.ci(res, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 3000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res, type = "norm")
##
## Intervals :
## Level      Normal
## 95%   (1011, 1150 )
## Calculations and Intervals on Original Scale
```